Prototype for the Simulation in Shared Virtual environments. Application to the supervision and the learning of the echographic gesture

C. Hamitouche[1], D. Gil[1], C. Roux[1], L. Ibañez[2], A. Plata[3], O.Gualdron[3]

[1] Image and Information Processing Department, LATIM, INSERM-ERM 0102;
Ecole Nationale Supérieure des Télécommunications de Bretagne, BP 832, 29285 BREST Cedex, FRANCE
[2] Division of NeuroSurgery, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27599, USA
[3] Escuela de Fisica, Universidad Industrial de Santander, A. A. 678 Bucaramanga, COLOMBIA

Abstract **This paper presents a prototype of a virtual shared environment in which several users can interact inside a Three Dimensional scene. Actions of a user are visible to others thanks to a protocol of transmission parameters of the scene. The communications are managed under a protocol of distributed objects, the visualization as well as the user interface use multi-platform libraries. This prototype is used to teach the echographic gesture.**
Keywords **Virtual environments, Virtual reality, Distributed Programming, echographic gesture learning**

I. INTRODUCTION

The current progress in the visualization field, virtual reality and telecommunications offers the possibility to develop systems of cooperative work with distant users. These systems provide new mechanisms of communication and interaction which replace textual explanations and explanatory diagrams.
The fundamental orientation of the prototype developed here is to provide a work environment in which experts cooperate with students to teach the manipulation of an echographic probe. The discussion is centred around a shared three-dimensional reconstruction of the studied body.

II. EVALUATION OF TOOLS

The first step in designing this prototype was an evaluation of existing tools suited for the application. The application can be viewed at three functionality levels :

- Communication
- Visualization
- User interface

First, several possibilities were analysed and tested :
- JAVA and VRML
- Java3D
- C ++, OpenGL, FLTK and Sockets

The final decision was in favour of the combination C++, OpenGL, FLTK and CORBA. Nevertheless, it is necessary to notice that some of these elements are in evolution and that future improvements can modify the compromises which are considered here.

A. JAVA and VRML

The first combination considered is that of *Virtual Reality Modelling Language* (VRML) with the language JAVA [1].
In this application a plugin which visualizes the VRML scene, and one or more Java Applets must be put on the same HTML page. These Java Applets communicate themselves with the VRML player to exchange information with the scene. VRML takes charge of the visualization layer, whereas Java can satisfy the needs of the communication and user interface.
However, drawbacks appear because of limitations imposed on applets for security reasons, more particularly concerning the access to files and to send messages between users. If these limitations can be removed, a Java Applet can make a communication with a distant server to have access to remote objects. In Java, the mechanism established to do this is known as *Remote Method Invocation (RMI*).
The RMI offers possibilities to the development of distributed applications. Many applications can be considered for Java programs which work as conventional applications in a virtual machine Java, instead of being applets working under HTML navigator.

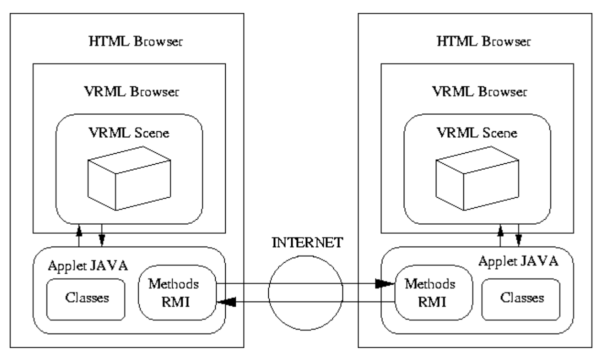

Figure 1. Configuration with VRML and JAVA. VRML supplies the services of visualization. Java Applets exchange the information with the navigator VRML and with the other users. The protocol of the RMI manages the connection between the users

B. JAVA3D

New versions of JAVA include multimedia packages oriented services. The visualization of 3D scenes is

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
25OCT2001	N/A	-

Title and Subtitle Prototype for the Simulation in Shared Virtual environments. Application to the supervision and the learning of the echographic gesture	**Contract Number** **Grant Number** **Program Element Number**
Author(s)	**Project Number** **Task Number** **Work Unit Number**
Performing Organization Name(s) and Address(es) Image and Information Processing Department, LATIM, INSERM-ERM 0102; Ecole Nationale Supérieure des Télécommunications de Bretagne, BP 832, 29285 BREST Cedex, FRANCE	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es) US Army Research, Development & Standardization Group (UK) PSC 802 Box 15 FPO AE 09499-1500	**Sponsor/Monitor's Acronym(s)** **Sponsor/Monitor's Report Number(s)**

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
Papers from the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, October 25-28, 2001, held in Istanbul, Turkey. See also ADM001351 for entire conference on cd-rom., The original document contains color images.

Abstract

Subject Terms

Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU

Number of Pages
4

managed thanks to the JAVA3D package [2]. This package represents one of the most modern approaches in terms of scene visualization oriented programming. It includes appropriate notions of the VRML and Quick-Time among other visualization systems.
JAVA3D calls upon the OpenGL layer services to make the visualization tasks. The intelligent management of the scene facilitate the optimization with respect to a program directly using OpenGL functions [3].
A hierarchy of objects is considered to accommodate the elements of the scene. Among them: 3D objects, lights, cameras, multiscale reference systems, and connections to make animations. VRML scenes can be loaded within this hierarchical structure. The class VRML Browser is included in Java3D. In this option, the visualization is performed by the package Java3D, communications by methods RMI, and the user interface by standard Java classes.
Complete applications for the shared virtual environments can be built with Java3D. Nevertheless, this platform is still in progress and does not represent a reliable support for real applications. Furthermore, limitations of the JAVA language license prevent considering it for the development of critical applications, more particularly in medicine.

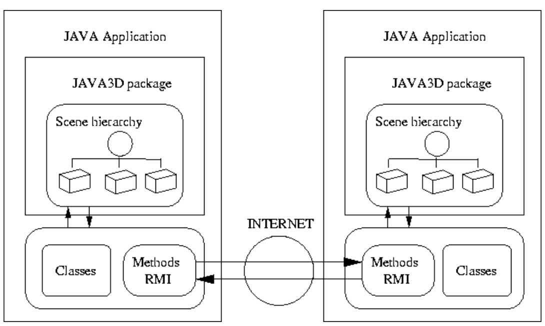

Figure 2. Java3D supplies the services of visualization by making Calls to the OpenGL library. VRML's notions are completely included in Java3D

C. C+ +, OpenGL, FLTK, Sockets

This combination can be described as the low level option since essential elements are used for each of the layers. The programming language is C++, the visualization is performed with OpenGL (or Mesa under Linux [4]), the user interface is operated with the library FLTK [5], and the communications managed with C++ classes which contain UNIX or Windows Sockets.

A maximum of flexibility is obtained but with maximal programming effort. Particularly, important limitations appear in the communications layer to manage all the events which can appear in the connection between users.
FLTK's choice for the user interface instead of the multiple options such as XForms, Motive, Qt, wxWindows, Gtk, or Tcl, was based on the fact of being the only library satisfying the following conditions :

- Source code available in licence GNU
- Oriented Object and interface in C++
- Interactive Construction of the interface and generation of the C ++ code.
- Portability with different platforms: (WindowsNT/95/98, Linux, Irix, Solaris,OS / 2)
- Interaction support with OpenGL
- Interaction support with CORBA

FLTK manages events by using sockets. It is so possible to insert the communications sockets inside the event management loop of the interface. This is important when programs are multithread.

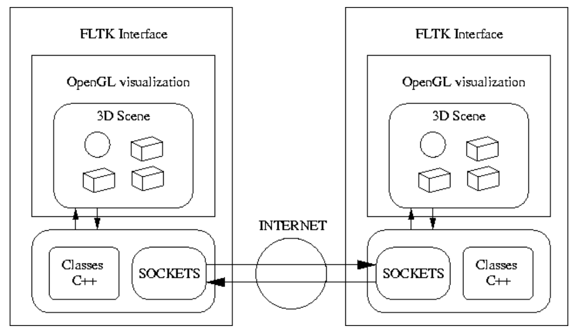

Figure 3. Configuration using sockets for Communications management, FLTK for the user interface and OpenGL for the visualization.

D. C ++ - OpenGL - FLTK – CORBA

In this platform, Sockets are replaced by CORBA[1] (Common Object Request Broker Structures) as low level elements for the management of the communications. Two implementations of CORBA are tested, MICO [6] and OmniORB [7], both exist in license GNU. CORBA offers a modern support for the distributed calculation. Objects on remote machines [8] [9] can be called to supply services.

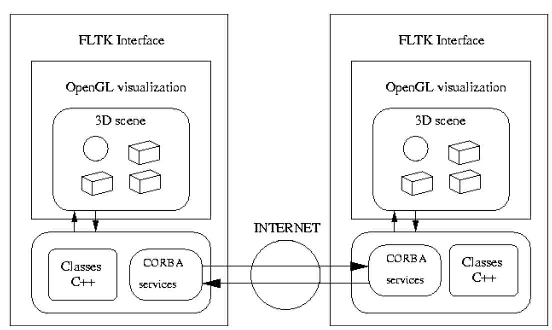

Figure 4. Platform using CORBA for the management of the communications, FLTK for the user interface and OpenGL for visualization

[1] http://www.corba.org

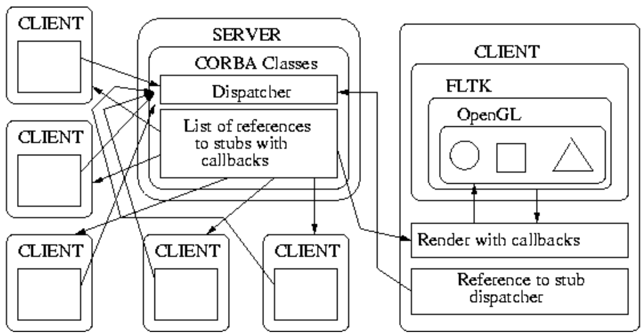

Figure 5. Diagram of the Client-Server configuration

III. APPLICATION

The configuration held for the development of the prototype is the combination of CORBA, OpenGL, FLTK and programming in C++. The aimed medical application is the learning of the manipulation of ultrasound probe. Echography is one of the many imaging modalities available in medicine for diagnosis, pathology follow up and interventional purposes. It provides images that are 2D slices of the 3D human body. These images contain morphological and functional information used in different medical fields such as, thrombus, echocardiography, obstetrics. Compared to the “ radiological ” modalities, it is specific with respect to its acquisition. The ultrasound (US) sensor used is manipulated by a physician according to the received images and to a mental strategy developed to fulfill some predefined objectives. In various clinical situations it is desirable to perform a distant supervision of a practitioner manipulating a US sensor.

The training scenario can be summarized as : The practitioner’s objective is to follow the location and attitude of the supervisor’s US sensor in order to have the same 2D US slice of the human body.

The supervisor transmits the actual US image, on which he reacts by moving the sensor in order to get a better slice. The supervised will move the local sensor on the actual patient in order to superimpose the transmitted slice and the actual one.

The teacher performs an echographic examination of an actual patient (slice by slice, or continuous scanning). The student reproduces on-line or off line the examination with a similar sensor in a simulation mode by superimposing the transmitted image and the local image.

The transmission/feedback loop can be performed on the basis of visual feedback using the shared virtual and collaborative environments mentioned above. During the training session, the student has the actual probe, and the teacher uses a Joystick.

The main objective is to reconstruct the 3D volume data of the body under study. A *Flock of Birds* localizer, using electromagnetic technology, is supplied with the probe in order to provide the 6 parameters of the location /attitude useful to 3D reconstruction of the volume data. The obtained volume data is the shared 3D scene.

The User interface is shown in figure 6 , where the upper part represents the login window, containing the FTP server address, login and password (for teacher and student). The 3D scene representing the volume data is located in the middle section, where the superimposed cutting planes of both the teacher and the student are shown, visual help consisting in the display of both 3D reference systems to be matched is integrated. The 2D images corresponding to the teacher and student cutting planes are continuously displayed. A communication window is also available for textual dialog between the teacher and the student.

This application allows the users to discuss elements of the scene by acting on the controls of the virtual probe to choose the best points of view.

IV. CONCLUSION

Within this paper is a recent investigations of the tools available for supporting the development of applications concerning the virtual shared environments. After an extensive evaluation of all possible tools and combinations, this investigation have lead to choose a set which satisfy our needs of multi-platform use.

A prototype has been developed taking into account the best combination of tools. This prototype has been used as support to an application for the supervision and the learning of the echographic gesture in a shared virtual environment.

ACKNOWLEDGMENT

This study is carried out within the framework of a program ECOS-Nord between France and Colombia.

V. REFERENCES

[1] Sams Teach Yourself Java 2 in 21 Days (Teach Yourself in 21 Days Series) by Laura Lemay, Rogers Cadenhead, Rev\&Updtd edition (April 1999) Sams

[2] Henry Sowizral, Kevin Rushforth, Michael Deering, *The Java 3d Api Specification (Java Series),* December 1997) Addison-Wesley Pub Co.

[3] *Opengl Programming Guide : The Official Guide to Learning Opengl, Version 1.1*, Mason Woo (Contributor), Jackie Neider, Tom Davis (Contributor), Opengl\H{ }Architecture Review boa, Paula Womack (Contributor), 2nd edition, Addison-Wesley Pub Co., January 1997.

[4] Brian Paul, *Mesa3D*, http://www.ssec.wisc.edu /~brianp/Mesa.html

[5] FLTK, *The Fast Light Tool Kit*, http://fltk.easysw.com /documentation.html
[6] Kay Romer and Arno Puder, *MICO is CORBA: CORBA 2.2 Implementation*, Universitat Frankfurt FB Informatik. International Computer Science Institute, Berkeley, USA.,1998,
[7] Sai-Lai Lo, David Riddoch, *The OmniORB2 version 2.7.1 User's Guide*, AT\&T Laboratories Cambridge. 22nd February1999, http://www.uk.research.att.com /omniORB/omniORB.html.
[8] OMG, *The Object Model*, http://www.infosys.tywien.ac.at/Research/Corba/OMG/obmod2.html
[9] Thomas J. Mowbray and Raphael C. Malveau, *CORBA design patterns*, John Wiley and Sons inc.,1997.

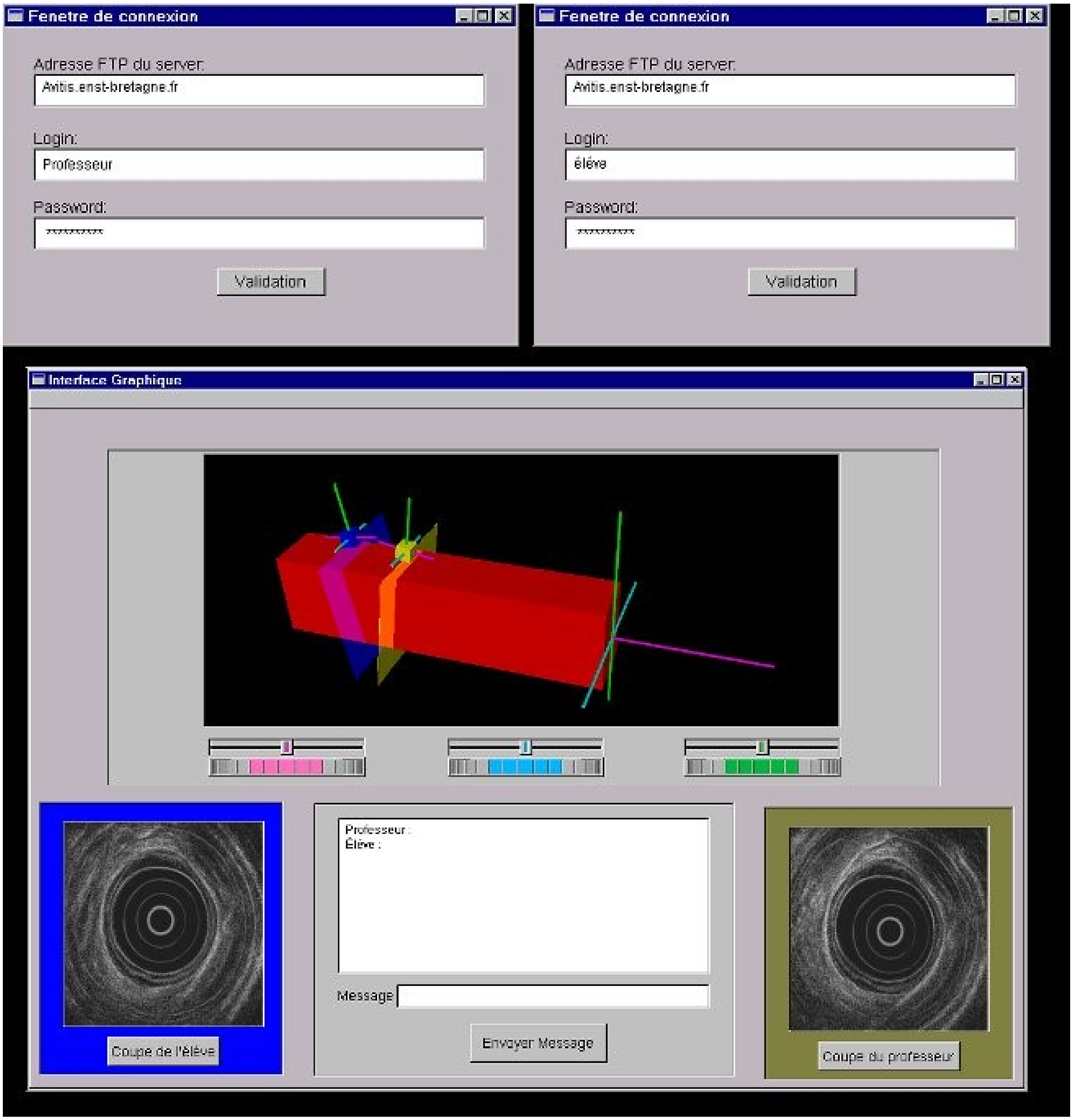

Figure 6 : User interface of the prototype for the supervision and the learning of the echographic gesture in Shared Virtual environments